

普通高等教育“十三五”计算机类规划教材

C YUYAN CHENGXU SHEJI
XIANGMUHUA JIAOCHENG

C语言程序设计 ——项目化教程

师 韵 王旭启 主编



北京邮电大学出版社
www.buptpress.com

普通高等教育“十三五”计算机类规划教材

C 语言程序设计

——项目化教程

主 编 师 韵 王旭启
副主编 范 晖 刘 影
张 玉 成 杜 刚



北京邮电大学出版社
www.buptpress.com

前 言

C 语言作为理工科学生必修的一门课程,体现了它作为高级语言在工科学生课程中的重要性。如何编写一本适合学生的好教材?好的教材能够引导学生循序渐进的学习知识,掌握知识,运用知识。在实际教学中,很多学生对学习 C 语言感到无从下手,觉得 C 语言学起来枯燥、难学。在教学过程中,总有学生问起学习 C 语言的最好方法是什么,答曰:读程序。既然读程序是学习 C 语言入门最快,也是最好的方法,为何不围绕着 C 语言的基础知识设计很多易于学生学习的例程呢?通过教学团队老师的细致分析,根据编者多年的一线教学实践经验,在教材中围绕知识点,设计了很多浅显易懂、紧扣知识点的实例。本书对每一个实例进行了深入分析,并在编译环境中进行调试、运行得到结果。这些实例都经过作者实践,真正做到了知识的由浅入深、由易到难。

本书共分为 10 个单元,下面对每个单元进行简单的介绍。

单元 1 主要介绍了计算机的发展过程,C 语言的运行环境,C 语言程序的基本结构。通过本单元学习,读者需要掌握 Visual C++ 6.0 的安装与使用,能编译、运行简单的 C 语言程序。

单元 2 主要讲解了 C 语言的标识符、关键字,基本数据类型,运算符及其表达式,数据类型转换等基础知识。在讲解这些知识时,提供了大量的案例,便于读者结合知识点在编译环境中练习、巩固学习的内容。

单元 3、单元 4 及单元 5 主要介绍了程序设计的三大结构:顺序结构、选择结构和循环结构的相关知识。

单元 6 主要讲解了数组的基本概念,数组的初始化及数组元素的引用,数据的查找、修改、排序等常用算法。

单元 7 主要讲解了函数的相关知识。函数是 C 语言程序的基本组成部分,学习函数的设计和调用是很重要的。本单元给出了很多经典的案例,比如水仙花数、汉诺塔、斐波那契数列等,以及实际应用的案例如出租车计费系统、学生成绩的统计与计算等。

单元 8 主要介绍了指针,主要包括指针的概念、指针与数组、指针与字符串等。

单元 9、单元 10 主要介绍了 C 语言的结构体、共用体以及文件操作,这些知识应用性强,通过相关的实例和项目设计,给读者真实的程序开发环境。

本书的特色主要体现在以下几个方面:

(1) 以“任务”为驱动,目标明确,案例通俗易懂,简单易学。通过任务描述、实例分析、实例实现等步骤,让读者一步一步体验学习 C 语言的乐趣。

(2) 项目设计更贴近真实软件开发环境。教材中设计的项目按照软件工程的思想,从需求分析、设计、编码、实现进行了细致的分析,带给读者更真实的软件开发过程。

(3) 举一反三的设计帮助读者步步提高学习 C 语言编程的能力,同时单元考核、线上练



习、答疑为读者提供全方位的答疑解惑,助力读者快速掌握 C 语言的开发方法。

为了读者可以了解 C 语言项目的开发过程,也便于今后走上工作岗位的需要,教材中还设计了很多的实际项目,比如电子计算器、出租车计费系统、通讯簿管理系统、学生成绩管理系统等,每个项目主要包括项目分析、项目设计、项目实现等。在每一单元后,设计了单元考核,考核题目紧紧围绕单元知识点,可以很好地巩固学习到的知识。读者在动手练习的过程中,一定会遇到很多的问题,建议多思考,借助于网络平台解决问题,并在问题解决后多总结。

本书第 1、6 单元及项目一由师韵编写,第 2、7 单元由王旭启编写,第 3、4 单元由刘影编写,第 5 单元由杜刚编写,第 8 单元及项目二由张玉成编写,第 9、10 单元及项目三由范晖编写,全书由师韵、王旭启统稿。

本书编写过程中,得到了许多人的帮助。全体成员在编写过程中,积极调研在学学生的学习感受和对 C 教材的期望,学生们提供了许多宝贵的意见,任课教师对教材也提供了很多有建设性的建议,在此一并感谢。还要感谢北京邮电大学出版社为作者出版本书提供机会和为编辑出版工作付出了辛勤劳动。

尽管在教材编写中,每一个编写人员都尽最大的努力,由于编者水平有限,教材中错误在所难免,恳请读者及同行、专家多提宝贵意见,我们将不胜感激。

编 者

原
书
缺
页



任务四	if 语句的嵌套	61
任务五	switch 语句	63
单元 5	循环结构	70
任务一	了解循环语句	70
任务二	while 语句的使用	72
任务三	do... while 语句的使用	73
任务四	for 语句的使用	75
任务五	多重循环	76
任务六	break 语句和 continue 语句的使用	77
任务七	程序举例	78
项目一	简易计算器项目实训	85
单元 6	数组	90
任务一	一维数组	90
任务二	二维数值数组	96
任务三	字符数组	100
任务四	字符串处理函数	103
任务五	冒泡排序	105
单元 7	函数	111
任务一	函数的基本概念	111
任务二	函数参数	115
任务三	函数调用	117
任务四	数组作为函数参数	121
任务五	函数的嵌套	123
任务六	函数递归	127
任务七	局部变量和全局变量	130
单元 8	指针	144
任务一	指针的基本概念	144
任务二	指针与一维数组	147
任务三	指针与二维数组	151
任务四	指针与字符串	154
任务五	函数传递值为指针类型	155
任务六	返回值类型为指针的函数	157
任务七	指针数组	158

项目二 通讯簿管理系统项目实训	166
单元 9 结构化数据	176
任务一 结构体数组的定义和使用	176
任务二 结构体指针的定义和使用	181
任务三 复杂结构体在函数中的使用	185
任务四 结构体、共用体和枚举的使用	190
单元 10 文件	200
任务一 文本文件的输入/输出	200
任务二 二进制文件的输入/输出	205
项目三 学生成绩管理系统软件项目实训	213
附录 A ASCII 码表	226
参考文献	227

单元 1 C 语言概述

随着“互联网+”时代的到来,计算机作为一种工具已经普遍地应用于各个领域。为了便于人们学习掌握计算机并且更好地与计算机交流,计算机科学家们经过不懈的努力,设计出了许多计算机语言,C语言就是其中之一,广泛地应用于工程计算领域,具有很强的数据处理能力。



学习任务:

- ◇ 计算机的发展过程。
- ◇ C 语言的运行环境。
- ◇ C 语言的基本结构。
- ◇ 算法。



学习目标:

- ◇ 掌握在 Visual C++6.0 中调试 C 语言程序的方法。
- ◇ 掌握 C 语言程序的基本结构。
- ◇ 掌握算法描述的方法。

任务一 计算机的发展过程



任务描述:

回顾计算机的发展简史,明确 C 语言在计算机体系中的位置。

人类所使用的计算工具是随着生产的发展和社会的进步,从简单到复杂、从低级到高级的发展过程,计算工具相继出现了如算盘、计算尺、手摇机械计算机、电动机械计算机等。

1946 年,世界上第一台电子数字计算机(ENIAC)在美国诞生。这台计算机共用了 18000 多个电子管组成,占地 170 m^2 ,总重量为 30 t,耗电 140 kW,运算速度达到每秒能进行 5000 次加法、300 次乘法。

从计算机的发展趋势看,大约 2010 年前美国就研制出千万亿次计算机。计算机在短短的 70 多年里经过了电子管、晶体管、集成电路(IC)和超大规模集成电路(VLSI)四个阶段的



发展,使计算机的体积越来越小,功能越来越强,价格越来越低,应用越来越广泛,目前正朝智能化(第五代)计算机方向发展。

按计算机语言的发展进程,可将计算机语言分为三类:

1. 机器语言

以二进制代码(0和1)表示机器指令的一种语言,其程序能被计算机直接执行。

2. 汇编语言

用助记符代替机器指令,用变量代替各类地址,称为汇编语言(也称符号语言)。

3. 高级语言

高级语言屏蔽了机器的细节,更接近于自然语言和数学语言,给编程带来了极大的方便。

C语言从发展进程来看,属于高级语言。

任务二 熟悉 C 程序的基本结构



任务描述:

在学习 C 语言程序设计之前,我们必须先熟悉一下 C 语言基本结构。

【实例 1】 编写程序,在屏幕上显示“欢迎来到 C 家族!”(这是一个 C 语言程序的基础实例)。

实例说明:

了解程序的结构和运行环境,通过输出函数输出字符串。运行结果如图 1-1 所示。



图 1-1 实例 1 运行结果

知识要点:

通过本实例的学习,让学生掌握 C 语言程序设计的基本结构,运行环境,调试方法。

1. C 语言程序的基本结构

下面先看两段程序代码。

代码一:该代码实现的功能是输出一条信息。

```
#include<stdio.h>           /* 文件包含 */
void main()                 /* 主函数 */
{                             /* 函数体开始 */
    printf("This is a C program.\n"); /* 输出语句 */
}                             /* 函数体结束 */
```



说明:

- (1) main—主函数名, void—函数类型;
- (2) 每个 C 语言程序必须有一个主函数 main ;
- (3) { }是函数开始和结束的标志,不可省;
- (4) 每个 C 语言语句以分号结束;
- (5) 使用标准库函数时应在程序开头一行写: #include <stdio. h>。

代码二:该代码的功能是求出两个数中的最大数。

```
#include "stdio. h"

void main ( )          /* 主函数 */
{
    int a, b, c;        /* 声明部分,定义变量 */
    scanf ( "% d, % d", &a, &b); /* 输入变量 a 和 b 的值 */
    c = max ( a, b);    /* 调用 max 函数,将得到的值赋给 c */
    printf ( "max = % d", c); /* 输出 c 的值 */
}

int max(int x, int y)  /* 定义 max 函数,函数值为整型,形式参数 x, y 为整型 */
{
    int z;             /* max 函数中的声明部分,定义本函数中用到的变量 z 为整型 */
    if (x > y) z = x;  /* if 语句,判断 x 大于 y 是否成立 */
    else z = y;
    return (z);       /* 将 z 的值返回,通过 max 带回调用处 */
}
```

说明:

本程序包括 main 和被调用函数 max 两个函数。max 函数的作用是将 x 和 y 中较大者的值赋给变量 z。return 语句将 z 的值返回给主调函数 main。

由以上两段程序代码可以看出,对于 C 语言程序:

(1) C 语言程序是由函数构成的,这使得程序容易实现模块化。

(2) 一个函数由两部分组成:

① 函数的首部:代码二中的 max 函数首部为 int max(int x, int y)。

② 函数体:花括号内的部分。若一个函数有多个花括号,则最外层的一对花括号为函数体的范围。

函数体包括两部分:

声明部分: int a, b, c。

执行部分:由若干个语句组成。

(3) C 语言程序总是从 main 函数开始执行的,与 main 函数的位置无关。

(4) C 语言程序书写格式自由,一行内可以写几个语句,一个语句可以分写在多行上,C 语言程序没有行号。

(5) 每个语句和数据声明的最后必须有一个分号。

(6) C 语言本身没有输入/输出语句。输入和输出的操作是由库函数 scanf() 和 printf() 等函数来完成的,C 语言对输入/输出实行“函数化”。



任务三 熟悉 C 语言的运行过程



任务描述：

当按照 C 语言的语法要求编写完程序后,计算机是不能直接运行的,必须对所编写的程序进行编译、连接,将其转换成可执行文件才可以运行。所以,在学习 C 语言程序设计之前,必须先熟悉一下 C 语言的编程环境。

知识要点：

高级语言编写的程序,不能被计算机直接执行,只有借助编译程序将其翻译为用 0 和 1 表示的机器语言指令代码,才能真正在计算机中执行。高级语言翻译有两种方式:一是编译方法,二是解释方法,C 语言采用编译方式。

编写好的 C 程序代码,需要经过上机输入及编辑源程序,对源程序进行编译,与库函数连接生成可执行目标文件之后才可运行。期间,把源程序代码称为源程序,扩展名为 .c;经过编译之后形成目标文件,扩展名为 .obj;经过连接之后形成的为可执行文件,扩展名为 .exe。图 1-2 给出了 C 语言程序的编辑、编译、连接和运行的过程。

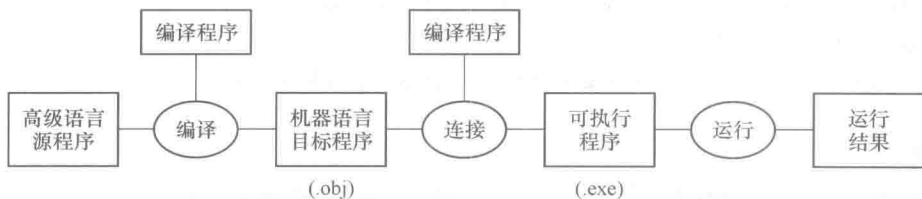


图 1-2 C 程序的编辑、编译、连接和运行的过程

1. 编写源代码

该阶段就是使用 C 语言编写实现特定功能的程序代码,通常将这一阶段的代码称为源代码(Source Code),源代码必须遵循 C 语言的规范。生成 .c 文件,一般称之为源文件。

2. 编译(Compile)

将编写的源代码翻译为计算机能够理解的二进制目标代码,由专门的编译器来完成。在编译阶段,编译器通常会发现源代码中的语法错误,程序员需要根据这些提示,进行修改,直至没有错误编译成功。生成 .obj 文件,称之为目标文件。

3. 连接(Link)

程序中除了自己编写的代码外,往往还需要调用库函数中的其他函数或者其他人编写的目标函数,连接过程便是将目标文件和函数合并生成完整的可执行文件,既生成 .exe 文件,这样程序就可以在计算机上运行了。

4. 运行

执行连接成功后的可执行文件,查验运行结果。



任务四 C 语言的开发环境



任务描述：

掌握 Microsoft Visual C++ 6.0 环境下的 C 语言程序开发过程。

Microsoft Visual C++ 系列是最经典的、功能强大的 C/C++ 开发工具，尤以 Visual C++ 6.0 (简称 VC 或者 VC 6.0) 版本使用最多。操作界面如图 1-3 所示。

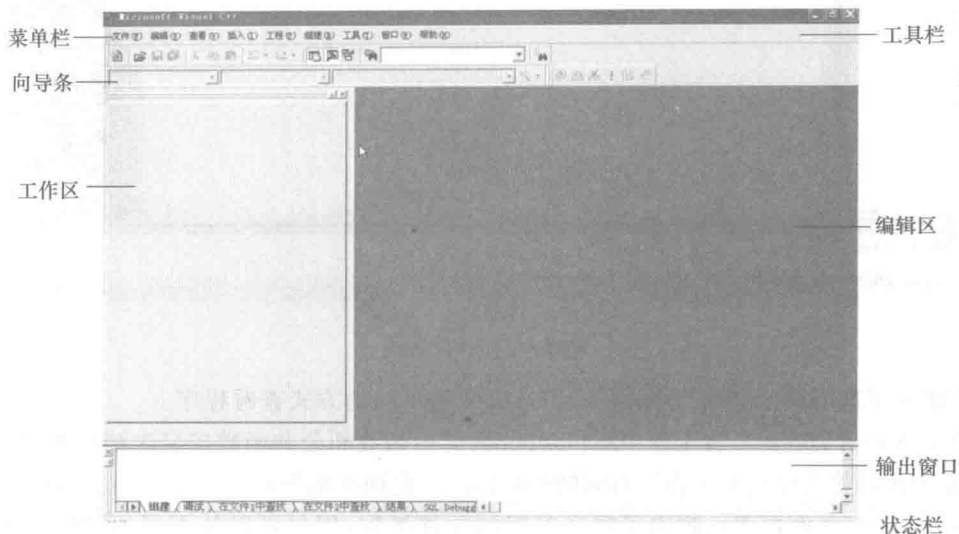


图 1-3 Visual C++ 6.0 操作界面

相关知识：

1. 菜单栏、工具栏和状态栏

Visual C++ 6.0 的窗口包含菜单栏、工具栏和状态栏，下面来分别介绍。

(1) 菜单栏。

- ① “文件”菜单：提供了文件和工程空间的新建、打开、保存等相关命令。
- ② “编辑”菜单：提供了复制、粘贴、查找等基本编辑命令，以及断点设置等命令。
- ③ “查看”菜单：主要用来打开和关闭各个功能性窗口。
- ④ “插入”菜单：用于将类、资源、文件、对象等添加到工程中。
- ⑤ “工程”菜单：用于设置工程属性、将工程添加到工作区。
- ⑥ “组建”菜单：用于程序的编译、连接、调试和运行。
- ⑦ “窗口”菜单：打开多个文件时，可改变窗口的显示方式，在各文件之间切换，或者关闭打开的窗口。

(2) 状态栏显示了程序的基本状态。

(3) 工具栏为常用命令的按钮形式。



2. 工作区

工作区窗口位于集成开发环境的左侧区域,该区域在 Visual C++ 6.0 刚启动时不显示任何内容。当加载或新建一个工作区文件(工程包含在其中)时,工作区内将显示当前工作区中文件的树形结构,如图 1-4 所示。

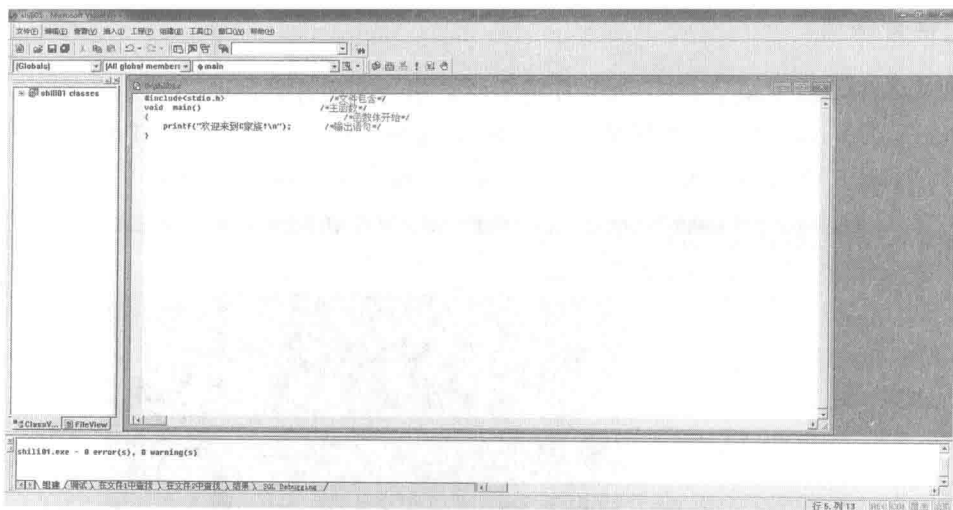


图 1-4 工作区界面

工作区下部有两个标签,分别允许用户以两种不同的方式查看程序。

Class View(类视图):将工程中所包含的类、类中函数和公共函数按层次结构列出。双击函数,则源代码将会出现在工作区右侧的编辑区内。此视图用于面向对象的 C++ 语言编程。

File View(文件视图):列出了整个工程的文件架构,用户可以从不同的文件夹中找到不同扩展名的文件,如源文件,头文件,资源文件等。

3. 编辑区

用户对代码和资源的一切操作都是在编辑区进行的。根据编辑的内容不同,编辑区会提供相应的功能。在编辑源程序时,编辑区是代码编辑窗口。

4. 输出窗口

输出窗口的作用是输出多种提示信息,主要包括编译程序的状态说明、出错信息、警告,以及某个变量的值或者某个关键字等。

任务五 第一个 C 语言程序



任务描述:

通过【实例 1】,掌握 C 语言程序在 Visual C++ 6.0 环境中的编辑、编译、连接、调试、运行的过程。



实现过程：

1. 创建源文件

步骤 1 启动 Visual C++ 6.0, 打开“文件”菜单, 选择“新建”菜单项, 在打开的“新建”对话框中选择“文件”选项卡, 在左侧文件类型列表中选择“C++ Source File”选项, 创建一个 C++ 源程序, 如图 1-5 所示。

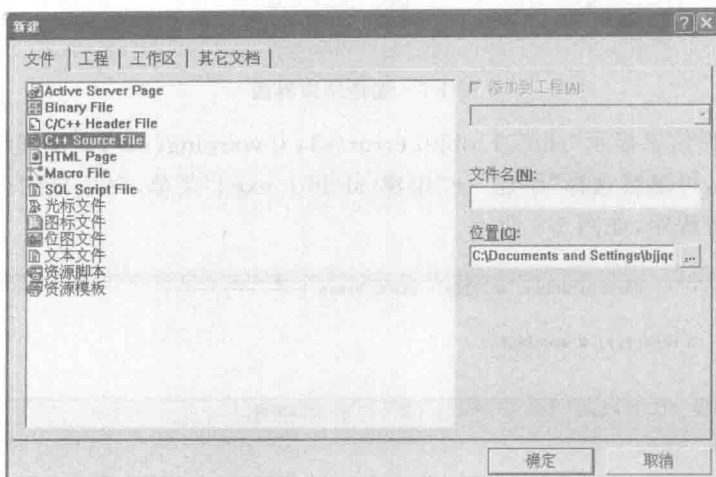


图 1-5 创建源文件

步骤 2 在右侧“文件名”编辑框中输入文件名“shili01.c”, 然后单击浏览按钮选择文件存储位置。单击“确定”按钮, 接下来在编辑区中输入以下代码:

```
#include<stdio.h>
main()
{
    printf("欢迎来到 C 家族! \n");
}
```

步骤 3 单击工具栏中的“保存”按钮或者直接按【Ctrl+S】组合键, 保存文件, 源程序创建完成。

2. 编译连接

步骤 1 选择“组建”→“编译[shili01.c]”菜单, 系统将显示如图 1-6 所示对话框, 询问是否在创建源文件的目录下建立一个活动工程和一个工作空间。

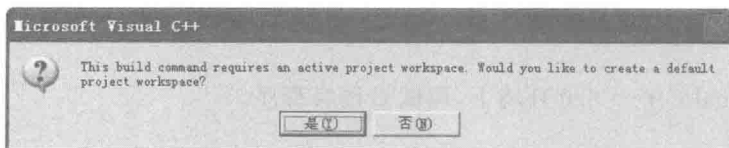


图 1-6 编译对话框

步骤 2 单击“是”按钮, 创建一个与源程序同名的工作区(对应文件为“shili01.dsw”)

和一个工程(对应文件为“shili01.dsp”),系统开始编译。编译结束后,将在输出窗口显示编译信息,如图 1-7 所示。

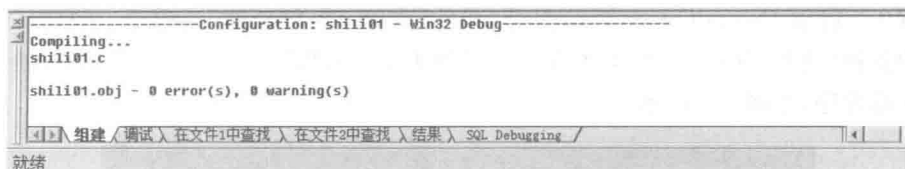


图 1-7 编译结束界面

步骤 3 编译信息显示“shili01.obj-0 error(s), 0 warning(s)”,表示编译程序时没有错误和警告。为此,可继续选择“组建”→“组建[shili01.exe]”菜单,对生成的目标程序进行连接,以生成可执行程序,如图 1-8 所示。

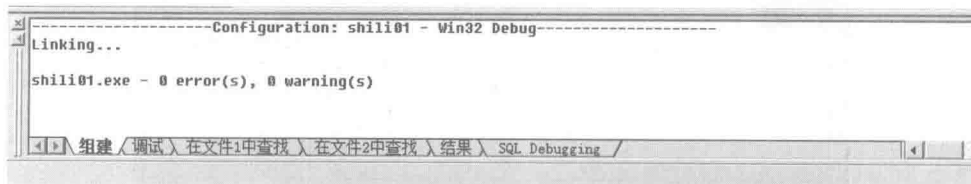


图 1-8 组建完成界面

3. 调试运行

编译连接通过后,选择“组建”→“执行[shili01.exe]”菜单,或者直接按【Ctrl+F5】组合键,运行生成的程序,将出现图 1-9 所示画面。结果正确无误,按任意键返回。



图 1-9 运行结果

任务六 Visual C++ 6.0 的调试功能





任务描述:

学习在 Visual C++ 6.0 环境下,调试 C 语言程序。

实现过程:

步骤 1 设置断点

程序成功编译后,将鼠标光标停留在需要设置断点的代码行,单击工具栏按钮即可添加断点,此时该行前端将出现一个断点标志,如图 1-10 所示。

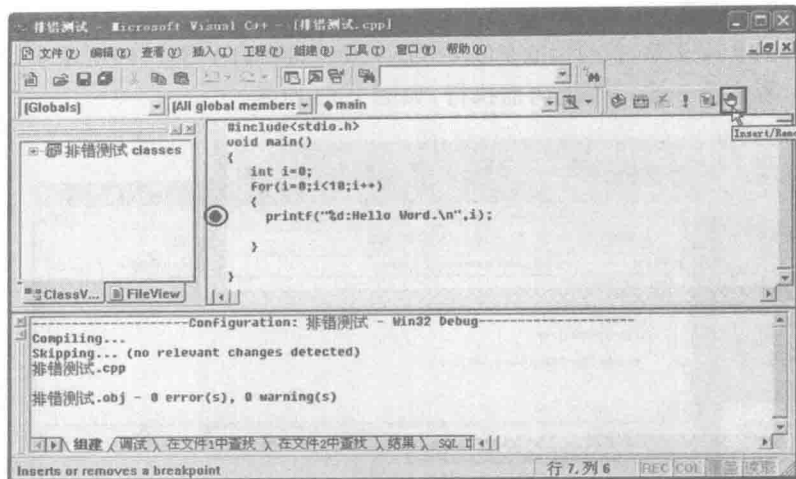


图 1-10 设置断点

步骤 2 开始调试

如图 1-11 所示,打开“组建”下拉菜单,执行“开始调试”→“GO”命令(或直接按【F5】键),程序会进入调试模式,并且会在断点处暂停,如图 1-12 所示。

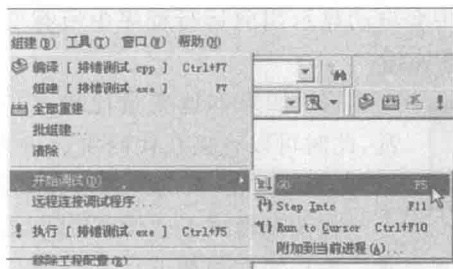


图 1-11 选择调试命令

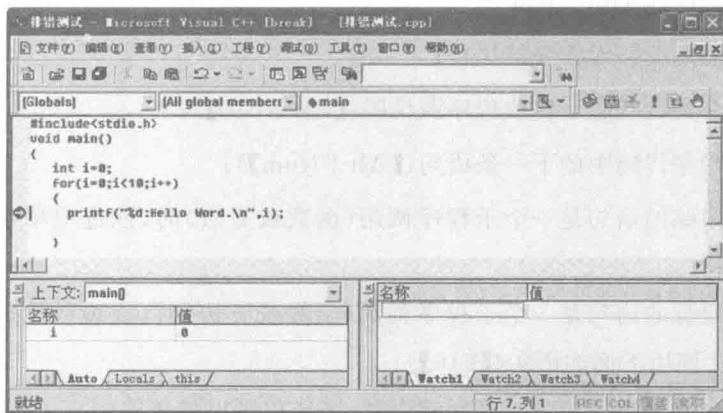


图 1-12 程序在断点处暂停

步骤 3 单步调试

打开“调试”下拉菜单,执行“Step Over”命令或直接按【F10】键,即可单步运行程序。不断按【F10】键,程序会一步一步地向前执行,如图 1-13 所示。

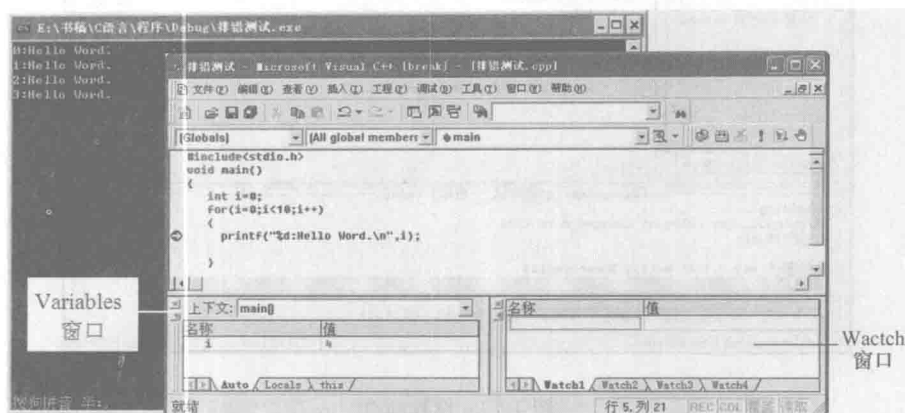


图 1-13 单步调试

单步调试程序时,可以在 Variables 窗口和 Watch 窗口中查看变量值的变化,这两个窗口的作用如下:

(1) 在 Variables 窗口中会自动显示当前运行程序中所有变量的值。随着单步调试的进行,会看到变量 i 的值逐渐递增。

(2) 如果本地变量比较多,Variables 窗口就会比较混乱,此时可以直接在代码中选中需要监控的变量,将其拖放到 Watch 列表,该变量的值会被显示出来。

在调试模式下,“调试”工具栏会自动弹出,如图 1-14 所示。各按钮作用如下:



图 1-14 “调试”工具栏

- ① 重启调试(【Ctrl+ Shift+F5】);
- ② 结束调试(【Shift+F5】);
- ③ 在当前点上挂起程序的执行;
- ④ 可以在调试状态下修改程序源代码(【Alt+F10】);
- ⑤ 显示程序代码中的下一条语句(【Alt+Num】);
- ⑥ 正在跟踪的语句是一个子程序调用(函数或方法)时,该选项单步进入所调用的子程序(【F11】);
- ⑦ 正在跟踪的语句是一个子程序调用(函数或方法)时,该选项跳过所调用的子程序,停留在子程序调用下面的语句(【F10】);
- ⑧ 确认当前子程序中没有任何程序错误时,该选项可以快速执行该子程序,并停留在子程序后面的语句(【Shift+F11】);
- ⑨ 快速执行到光标所在的代码处(【Ctrl+F10】);